date
exp no 16a 8. Implement a LSTM

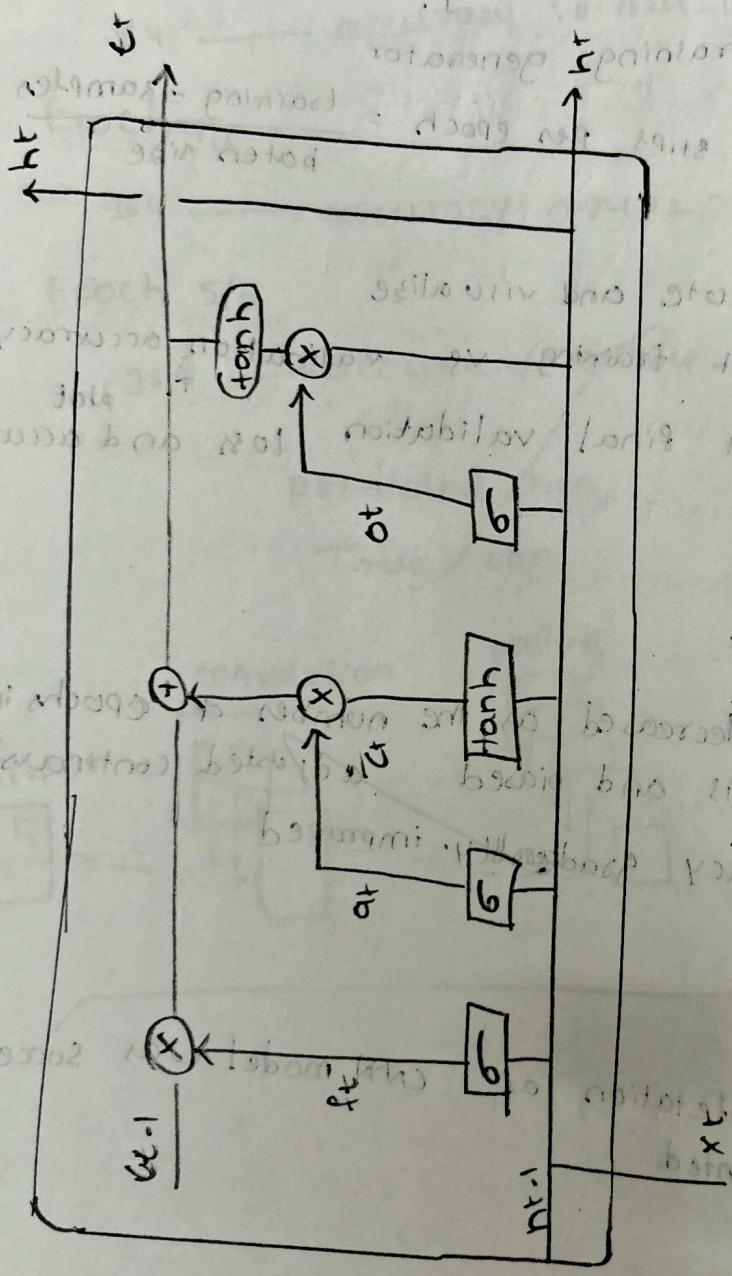
Aim:

To implement a long short term model using Pytorch for time series forecasting.

Algorithm:-

* Data preparation - generate a sine wave data set as the time series input
* for each time step in the input sequence
  - compute forget gate to decide what information to discard.
  - compute input gate to decide what new information to add
  - update cell state combining old state and new candidate values.
* use the final hidden state to predict the next value in the sequence
* train the network using backpropagation through (BPTT) and update parameters to minimize

LSTM internal architecture



Pseudo code:

input: sequence $x = [x_1, x_2, \ldots, x_T]$
output: predicted next value y_hat

initialize parameters: W_f, W_i, W_c, W_o, b_f, b_i, b_c, b_o.
initialize states: h_0, c_0

for t = 1 to T do

$f_t = \text{sigmoid}(W_f * \text{concat}(h_{t-1}, x_t) + b_f)$

$i_t = \text{sigmoid}(W_i * \text{concat}(h_{t-1}, x_t) + b_i)$

$c\_tilde_t = \tanh(W_c * \text{concat}(h_{t-1}, x_t) + b$

$c_t = f_t * c_{t-1} + i_t * c\_tilde_t$

$o_t = \text{sigmoid}(W_o * \text{concat}(h_{t-1}, x_t) + b_o$

$h_t = o_t * \tanh(c_t)$

end for

y_hat = Dense(h_t).

Return y_hat

Result: The model successfully learns the pattern and predicts the next value close to the expected trend.

Epoch 1/5
val-accuracy: 0.8462 , val_loss: 0.3826

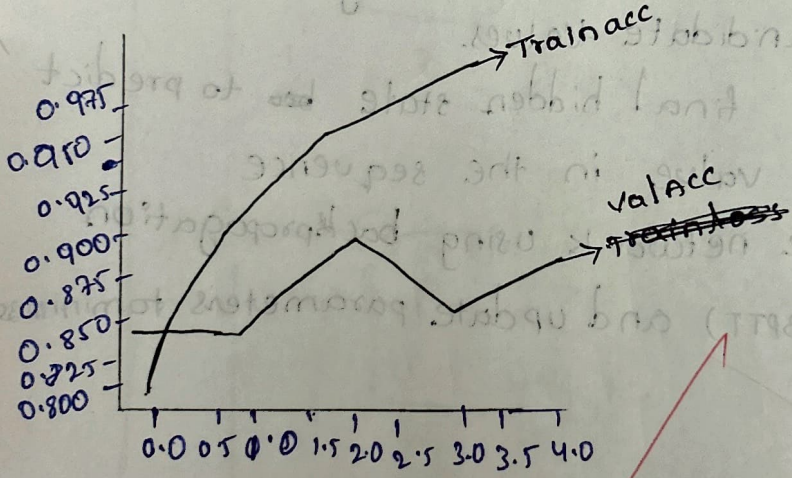
Epoch 2/5
val-accuracy: 0.8472 , val_loss: 0.3478

Epoch 3/5
val_accuracy: 0.8682 , val_loss: 0.344,

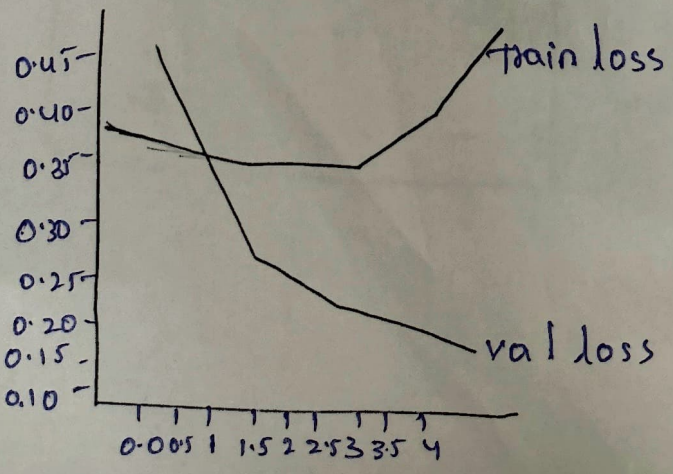
Epoch 4/5
val_accuracy: ~~0.8611~~ 0.8532 - val loss: 0.3

Epoch 5/5
val_accuracy: 0.8618 · val loss: 0.4712

Accuracy



loss



```python
# LSTM Implementation
import tensorflow as tf
from tensorflow.keras.datasets import imdb
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout
import matplotlib.pyplot as plt

# Load IMDB dataset
vocab_size = 10000
maxlen = 200

(x_train, y_train), (x_test, y_test) = imdb.load_data(num_words=vocab_size)
x_train = pad_sequences(x_train, maxlen=maxlen)
x_test = pad_sequences(x_test, maxlen=maxlen)

# Build LSTM model
lstm_model = Sequential([
    Embedding(input_dim=vocab_size, output_dim=128, input_length=maxlen),
    LSTM(128, dropout=0.2, recurrent_dropout=0.2),
    Dense(1, activation='sigmoid')
])

lstm_model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Train model
lstm_history = lstm_model.fit(
    x_train, y_train,
    epochs=5,
    batch_size=64,
    validation_split=0.2
)

# Evaluate model
lstm_score = lstm_model.evaluate(x_test, y_test)
print(f"LSTM Test Accuracy: {lstm_score[1]*100:.2f}% | Loss: {lstm_score[0]:.4f}")

# Plot Accuracy and Loss
plt.figure(figsize=(12,5))

# Accuracy
plt.subplot(1,2,1)
plt.plot(lstm_history.history['accuracy'], label='Train Accuracy')
plt.plot(lstm_history.history['val_accuracy'], label='Validation Accuracy')
plt.title('LSTM Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()

# Loss
plt.subplot(1,2,2)
plt.plot(lstm_history.history['loss'], label='Train Loss')
plt.plot(lstm_history.history['val_loss'], label='Validation Loss')
plt.title('LSTM Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.show()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/imdb.npz
17464789/17464789 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Epoch 1/5
/usr/local/lib/python3.12/dist-packages/keras/src/layers/core/embedding.py:97: UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
313/313 ━━━━━━━━━━━━━━━━━━━━ 134s 403ms/step - accuracy: 0.6963 - loss: 0.5653 - val_accuracy: 0.8452 - val_loss: 0.3865
Epoch 2/5
313/313 ━━━━━━━━━━━━━━━━━━━━ 136s 389ms/step - accuracy: 0.8551 - loss: 0.3442 - val_accuracy: 0.8556 - val_loss: 0.3618
Epoch 3/5
313/313 ━━━━━━━━━━━━━━━━━━━━ 139s 379ms/step - accuracy: 0.8921 - loss: 0.2707 - val_accuracy: 0.8488 - val_loss: 0.3664
Epoch 4/5
313/313 ━━━━━━━━━━━━━━━━━━━━ 141s 377ms/step - accuracy: 0.9095 - loss: 0.2346 - val_accuracy: 0.8410 - val_loss: 0.3953
Epoch 5/5
313/313 ━━━━━━━━━━━━━━━━━━━━ 118s 378ms/step - accuracy: 0.9335 - loss: 0.1808 - val_accuracy: 0.8434 - val_loss: 0.3769
782/782 ━━━━━━━━━━━━━━━━━━━━ 38s 48ms/step - accuracy: 0.8463 - loss: 0.3841
LSTM Test Accuracy: 84.69% | Loss: 0.3827
```